UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/004,196	11/14/2001	Jose Fernandez	42390P10728	5794

8791 7590 01/22/2007
BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
GUILL, RUSSELL L	
ART UNIT	PAPER NUMBER
2123	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/22/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/004,196	**Applicant(s)** FERNANDEZ, JOSE	
	Examiner Russ Guill	**Art Unit** 2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 December 2006*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5 and 7-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5 and 7-49* is/are rejected.

7)☒ Claim(s) *36* is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 December 2005* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This action is in response to an Amendment filed December 4, 2006. No claims have been added or cancelled. Claims 1 - 5 and 7 - 49 are pending. Claims 1 - 5 and 7 - 49 have been examined. Claims 1 - 5 and 7 - 49 have been rejected.

2. **As previously recited, the Examiner would like to thank the Applicant for the well-presented amendment, which was useful in the examination process. The Examiner appreciates the effort to carefully analyze the Office Action and make appropriate arguments and amendments.**

3. This action is NON-final due to new rejections made under 35 U.S.C. § 101.

Response to Arguments

4. Regarding **claims 1 - 8** rejected under 35 U.S.C. § 112:

4.1. Applicant's amendments to the claims overcome the rejection.

5. Regarding **claims 29 - 35** rejected under 35 U.S.C. § 101:

5.1. Applicant's amendments to the claims overcome the rejection.

6. Regarding **claims 1 - 3, 7 - 13, 15 - 20 and 22 - 23** rejected under 35 U.S.C. § 103:

6.1. Applicant's arguments and claim amendments have been fully considered, but are not persuasive. The Examiner respectfully maintains that there is no difference between the XML documents and the persistent data as claimed. The details are provided in the rejection below.

Claim Objections

7. Claim 36 is objected to for the following minor informalities: the first limitation is terminated with a pair of semicolons, where only one is needed.

Claim rejections 35 USC § 112

8. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

9. **Claims 1 - 8, 9 - 28, 36 - 42 and 43 - 49** are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

9.1. Claims 1, 9, 36 and 43 recite, "metadata describing the persistent data including a class file with bytecodes for each object". A class file does not appear to be described in the disclosure.

9.2. Claims **2 - 8, 10 - 28, 37 - 42 and 44 - 49** are rejected based on their dependency on their respective intermediate and parent claims which are rejected under 35 U.S.C. 112.

Claim Rejections - 35 USC § 101

10. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10.1. **Claims 9 - 28, 29 - 35, 36 - 42, 43 and 45 - 49** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

a. Regarding **claims 9 - 28**, the recited apparatus appears to contain software, which appears to be an abstract idea. Therefore, to be statutory, the claim must be directed to a practical application having a concrete, useful and tangible result. The claims do not appear to produce a tangible result needed to support a practical application. In addition, the receiving system appears to be entirely software, but there is no processor that is operationally coupled to the software to realize the functionality of the software, and therefore, the claim cannot have a tangible result. In order to expedite the examination process, please note that there are two separate issues: first, the claims do not have any limitations that appear to produce a tangible result, and second, a processor is also needed to realize the tangible result. Further, please note that a functional limitation will usually not produce a tangible result.

b. Regarding **claims 29 - 35**, the recited apparatus appears to contain software, which appears to be an abstract idea. Therefore, to be statutory, the claim must be directed to a practical application having a concrete, useful and tangible result. The claims do not appear to produce a tangible result needed to support a practical application. In addition, the apparatus does not appear to have processor that is operationally coupled to the software to realize the functionality of the software, and therefore, the claim cannot have a tangible result.

c. Regarding **claims 36 - 42**, the recited method appears to contain abstract operations such as comparing a data model description to a preexisting data model. Therefore, to be statutory, the claim must be directed to a practical application having a concrete, useful and tangible result. The claims do not appear to produce a tangible result needed to support a practical application.

d. Regarding **claims 43 and 45 - 49**, the recited article of manufacture appears to contain abstract operations such as establishing a storage format. Therefore, to be statutory, the claim must be directed to a practical application having a concrete, useful and tangible result. The claims do not appear to produce a tangible result needed to support a practical application.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning (U.S. Patent Application Publication Number US 2002/0103829) in view of Morgenstern (U.S. Patent No. 5,970,490).

12.1. Regarding **claim 1**, Manning appears to teach:

12.1.1. receiving a persistence package (*figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004]; and paragraph [0028]; please note that the DTD is included with the received document*) from one of a plurality of different software components (*paragraph [0004] first and second sentences; it would have been obvious that the multiple data types recited, such as vector graphics and e-commerce transactions, were produced by different software components, wherein a software component is interpreted to include an application*), the persistence package including persistent data and metadata (*figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004]; and paragraph [0028]; please note that the DTD is included with the received document*), the software components having persistent data in different formats (*paragraph [0004] first and second sentences; paraphrasing a definition of "format" from the IBM Dictionary of Computing, a "format" is a specified arrangement of fields; it would have been obvious that persistent data from vector graphics is in a different format than e-commerce transactions*).

12.1.2. extracting persistent data and metadata from the persistence package (*figure 3, items 114 - 128; and paragraphs [0028] and [0029]*), the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component (*paragraph [0004]*), the metadata describing the persistent data (*paragraph [0004]*) including a class file with bytecodes for each object (*paragraphs [0004] - [0005], it would have been obvious that a DTD was a class file with bytecodes for each document object*).

12.1.3. establishing, based on the extracted metadata, a storage format for the persistent data during a runtime of a receiving system (*paragraph [0028]; paragraph [0041]; figure 3, elements 102 - 110*).

12.1.4. applying the transform to the persistent data to format the persistent data without using the software component from which the persistence package was received during the runtime of the receiving system (*figure 3, element 124, since the accessed object is stored, it would have been obvious that the transform is applied; and paragraph [0029], since each object (e.g. attribute value or content) is stored in an element table, it would have been obvious that a transform is applied; please note that the specification recites, "Transforms establish a storage format and/or storage location for the persistent data."*) from the format of the software component into a storage format that is compatible with the receiving system and with a storage device independent of the software component (*paragraph [0022], paragraphs [0027] - [0029], and paragraph [0034]; it would have been obvious that the received data such as vector graphics and e-commerce transactions is being stored in a relational database, which is a storage format that is compatible with the receiving system and with a storage device independent of the software component*).

12.1.5. storing the persistent data in the storage device in the storage format during the runtime of the system (*paragraph [0029]; and figure 3, element 124*).

12.1.6. Manning does not specifically teach:

12.1.7. establishing, based on the extracted metadata, *a transform for* a storage format for the persistent data during a runtime of the receiving system.

12.1.8. <u>Morgenstern appears to teach</u>:

12.1.9. establishing, based on the extracted metadata, <u>*a transform for*</u> a storage format for persistent data (<u>*figure 2, elements 42, 46, 22, 32, 36, 23, 66; and column 8, lines 53 - 67, and column 9, lines 1 - 3, and column 7 lines 16 - 67, and column 8, lines 1 - 53*</u>).

12.1.10. The motivation to use the art of Morgenstern with the art of Manning would have been the several benefits recited in Morgenstern, including that self-description information simplifies the management of generated source code and the resulting compiled modules, which is especially useful in large systems (<u>*column 6, lines 33 - 37, lines 1 - 2, lines 9 - 12*</u>), and the advantage (<u>*column 46, lines 41 - 45*</u>) that the data transformation approach allows rules to be more declarative in nature, and also supports asynchronous processing of transformations, thereby being amenable to parallelization (<u>*column 46, lines 35 - 41*</u>), which would have been recognized as an advantage by the ordinary artisan.

12.1.11. Therefore, as discussed above, it would have been obvious to the ordinary artisan to use the art of Morgenstern with the art of Manning to produce the claimed invention.

12.2. Regarding claim 2, Manning appears to teach using metadata passed from the persistence package to establish a storage location for the persistent data during the runtime of the system <u>*(paragraphs [0028] - [0029]; Since the Applicant's specification provides no detail on the meaning of "storage location", the term "storage location" is being given a broad reasonable interpretation to include database tables as a storage location)*</u>.

12.3. Regarding claim 3, Manning appears to teach that the metadata comprises at least in part a description of a model structure of the persistent data *(figure 3, item 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004])*.

12.4. Regarding claim 7, Manning appears to teach retrieving persistent data from storage using a transform during the runtime of the system *(figure 4, all items; and paragraph [0030])*.

12.5. Regarding claim 8, Manning appears to teach receiving persistent data compatible with at least one of any type of processor, any type of programming language, any type of operating system, and any type of architecture *(paragraph [0021])*.

12.6. Regarding claim 9, Manning appears to teach:

12.6.1. Almost all of claim 9 is taught as described in claim 1 above. The differences are taught below.

12.6.2. a data storage device *(paragraph [0021])*.

12.7. Regarding claim 10, Manning appears to teach the data storage device is external to a receiving system using the persistence engine *(paragraph [0021])*.

12.8. Regarding claim 11, Manning appears to teach a storing interface to store the persistent data using the storage format *(paragraph [0027], second sentence])*.

12.9. Regarding claim 12, Manning appears to teach a retrieving interface to retrieve the persistent data for use by one of the receiving system and the software component, the software component comprising an application *(figure 1, element 2; and paragraph [0011]; and paragraph [0027], second sentence; and paragraph [0030])*.

12.9.1. Regarding *(figure 1, element 2; and paragraph [0011]; and paragraph [0027], second sentence; and paragraph [0030])*; it would have been obvious to the ordinary artisan that the received query is from one of a running system and a software component, the software component comprising an application.

12.10. Regarding claim 13, Manning appears to teach that the metadata comprises at least in part a description of the data model structure of the persistent data *(figure 3, item 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004])*.

12.11. Regarding claim 15, Manning appears to teach that the persistence engine receives a persistence package comprising the metadata and the persistent data *(figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004]; and paragraph [0028])*

12.12. Regarding claim 16, Manning appears to teach that the persistence engine receives persistent data structured using any data model from a source comprising at least one of any type of processor, any type of operating system, any type of programming language, and any type of architecture *(figure 3; all items)*.

12.12.1. Regarding *(figure 3; all items)*; it was inherent that any type of data model can be expressed in an XML document.

12.13. Regarding claim 17, Manning appears to teach a metadata engine having a metadata reader *(paragraph [0027] - please note that it was inherent that the XML document manager includes a metadata reader)* and a metadata filter *(paragraph [0027] - please note that the XML parser was a metadata filter)*.

12.14. Regarding claim 18, Manning appears to teach that the metadata filter interprets the metadata *(paragraph [0027])*.

12.15. Regarding claim 19, Manning appears to teach a transform engine having a set of transforms, a transform selector, and a transform generator *(figure 3, items 102 - 110; and paragraph [0028])*.

12.16. Regarding claim 20, Manning appears to teach that a transform establishes at least one of the storage format and the storage location to store the persistent data in the data storage device *(paragraphs [0028] and [0029])*.

12.17. Regarding claim 22, Manning appears to teach that a transform selector selects a transform based on filtered metadata *(figure 3, items 100 - 102; and paragraphs [0027] and [0028])*.

12.18. Regarding claim 23, Manning appears to teach that a transform selector requests a transform from the transform generator based on filtered metadata *(figure 3, items 100 - 110; and paragraphs [0027] and [0028])*.

==

13. **Claims 4 - 5 and 14** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23** above, further in view of XML ("Extensible Markup Language (XML) 1.0"; W3C Recommendation 10-Feb-98, 1998).

13.1. Manning as modified by Morgenstern teach the method and apparatus of a persistence engine as recited in **claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23** above.

13.2. Claim 4 is a dependent claim of claim 3, and thereby inherits all of the rejected limitations of claim 3.

13.3. Claim 5 is a dependent claim of claim 4, and thereby inherits all of the rejected limitations of claim 4.

13.4. Claim 14 is a dependent claim of claim 13, and thereby inherits all of the rejected limitations of claim 13.

13.5. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)*.

13.6. The art of XML is directed toward describing the Extensible Markup Language (XML) *(Abstract)*.

13.7. Regarding claim 5, Manning appears to teach that extracting persistent data and metadata from a persistence package comprises using a filter *(paragraph [0027] - please note that the XML parser is a filter; and figure 3, all items; and paragraphs [0028] and [0029])*.

13.8. Regarding claim 4, Manning as modified by Morgenstern does not specifically teach that the metadata conforms to a metadata template comprising rules for describing the model structure.

13.9. Regarding claim 14, Manning as modified by Morgenstern does not specifically teach a metadata template to format the metadata for readable reception by the persistence engine.

13.10. Regarding claim 4, XML appears to teach that the metadata conforms to a metadata template comprising rules for describing the model structure *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*.

13.10.1. Regarding *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*; the reference XML describes the rules that the metadata conforms to.

13.11. Regarding claim 14, XML appears to teach a metadata template to format the metadata for readable reception by the persistence engine *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*.

13.11.1. Regarding *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*; the reference XML describes the rules that the metadata conforms to, and specifically the production in section 2.1 is a metadata template.

13.12. The art of XML and the art of Manning as modified by Morgenstern are analogous art because they both contain the art of interpreting XML documents.

13.13. The motivation to combine the art of XML with the art of Manning and Morgenstern would have been obvious given the need in Manning to interpret XML documents, and the rules given in XML to form valid XML documents. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of XML with the art of Manning and Morgenstern to produce the claimed invention.

14. **Claim 21** is rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to claims **1 - 3, 7 - 13, 15 - 20, and 22 - 23** above, further in view of DeltaXML (web page for DeltaXML.com from September 2001 using www.archive.org at web.archive.org/web/20011021144026/www.deltaxml.com/prod-xmlschema-1000.html).

14.1. Manning as modified by Morgenstern teaches the persistence engine as recited in claims **1 - 3, 7 - 13, 15 - 20, and 22 - 23** above.

14.2. Claim 21 is a dependent claim of claim 19, and thereby inherits all of the rejected limitations of claim 19.

14.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Title and Abstract; and paragraph [0020] regarding the XML document)*:

14.4. The art of DeltaXML is directed toward comparing XML schema DTD files to determine differences *(page 1, box labeled Description)*.

14.5. Regarding claim 21, Manning as modified by Morgenstern does not specifically teach that the transform selector comprises a data model comparator.

14.6. DeltaXML teaches a data model comparator *(page 1, box labeled Description)*, which also calculates the data model variance.

14.7. The art of DeltaXML and the art of Manning as modified by Morgenstern are analogous art because they both contain the problem of determining whether a pair of DTD's are different *(Manning, lines 14 - 17 of paragraph [0028])*.

14.8. The motivation to use the art of DeltaXML with the art of Manning and Morgenstern would have been obvious given the need recited in Manning to determine whether documents have different DTD's *(Manning, lines 14 - 17 of paragraph [0028])*. Therefore, as discussed above, it would have been obvious to use the art of DeltaXML with the art of Manning and Morgenstern to produce the claimed invention.

15. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23 above, further in view of Kanne (Kanne, Carl-Christian; Moerkotte, Guido; "Efficient storage of XML data", 1999, Technical Report 8/99, University of Mannheim).

15.1. Manning as modified by Morgenstern teaches the persistence engine as recited in claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23 above.

15.2. Claim 25 is a dependent claim of claim 23, and thereby inherits all of the rejected limitations of claim 23.

15.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)*.

15.4. The art of Kanne is directed toward efficient storage of XML data *(Title)*.

15.5. Manning as modified by Morgenstern does not specifically teach that the transform generator produces a transform that substantially maintains the model structure of the persistent data received by the receiving system.

15.6. Kanne appears to teach that the transform generator produces a transform that substantially maintains the model structure of the persistent data received by the receiving system *(pages 4 - 5, section 2.2 Logical Model - please note the use of a tree structure for XML. XML was inherently tree structured.)*.

15.7. The art of Kanne and the art of Manning as modified by Morgenstern are analogous art because they are both directed to the storage of XML data.

15.8. The motivation to use the art of Kanne with the art of Manning as modified by Morgenstern would have been obvious given the benefit recited in Kanne of describing a method to dynamically maintain efficient physical storage for large tree structured objects *(page 20, section 6 Conclusion and Future Work)*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Kanne with the art of Manning as modified by Morgenstern to produce the claimed invention.

16. Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to claims **1 - 3, 7 - 13, 15 - 20, and 22 - 23** above, in view of Schoning (Schoning, Harald; "Tamino - a DBMS Designed for XML", 2001 Proceedings 17th International Conference on Data Engineering, 2-6 April 2001).

16.1. Manning as modified by Morgenstern teaches the persistence engine as recited in **claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23** above.

16.2. Claim 26 is a dependent claim of claim 23, and thereby inherits all of the rejected limitations of claim 23.

16.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Title and Abstract; and paragraph [0020] regarding the XML document)*.

16.4. The art of Schoning is directed toward a database management system designed for XML *(Title)*.

16.5. Manning as modified by Morgenstern does not specifically teach that the transform generator produces a transform to remodel the persistent data to maximize efficient retrieval for an application.

16.6. Regarding claim 26, Schoning appears to teach that the transform generator produces a transform to remodel the persistent data to maximize efficient retrieval for an application *(page 152, section labeled "Indexing and storage methods")*.

16.6.1. Regarding *(page 152, section labeled "Indexing and storage methods")*; it would have been obvious to design the transform generator to produce a transform to remodel the persistent data to maximize efficient retrieval for an application.

16.7. The art of Manning as modified by Morgenstern and the art of Schoning are analogous art because they are both directed to the art of XML databases.

16.8. The motivation to use the art of Schoning with the art of Manning as modified by Morgenstern would have been obvious given the statement recited in Schoning that indexes are indispensable in database systems because otherwise large amounts of data could not be efficiently queried *(page 152, section labeled "Indexing and storage methods")*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Schoning with the art of Manning as modified by Morgenstern to produce the claimed invention.

17. **Claim 27** is rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23** above, in view of Ives (Ives, Zachary G.; Florescu, Daniela; Friedman, Marc; Levy, Alon; Weld, Daniel S.; "An Adaptive Query Execution System for Data Integration", 1999, SIGMOD 1999).

17.1. Manning as modified by Morgenstern teaches the apparatus of a persistence engine as recited in **claims 1 - 3, 7 - 13, 15 - 20, and 22 - 23** above.

17.2. Claim 27 is a dependent claim of claim 23, and thereby inherits all of the rejected limitations of claim 23.

17.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Title and Abstract; and paragraph [0020] regarding the XML document)*.

17.4. The art of Ives is directed toward an adaptive query execution system for data integration *(Title)*.

17.5. Regarding claim 27, Manning as modified by Morgenstern does not specifically teach that the transform generator uses iterative read-write trials to produce a transform to remodel the persistent data to maximize storage and/or retrieval speed.

17.6. Regarding claim 27, Ives appears to teach that the transform generator uses iterative read-write trials to produce a transform to remodel the persistent data to maximize storage and/or retrieval speed *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*.

17.6.1. Regarding *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*; it would have been obvious to design the transform generator to use iterative read-write trials to produce a transform to remodel the persistent data to maximize storage and/or retrieval speed.

17.7. The art of Manning as modified by Morgenstern and the art of Ives are analogous art because they are both contain the problem of data queries *(Manning, paragraph [0030])* and Ives *(Title)*.

17.8. The motivation to use the art of Ives with the art of Manning as modified by Morgenstern would have been obvious given the statement recited in Ives that it is important to optimize the time to initial answers to a query *(page 300, left-side column, the paragraph that starts with "Since data integration . . . ")*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Ives with the art of Manning as modified by Morgenstern to produce the claimed invention.

18. **Claims 24 and 28** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to claims **1 - 3, 7 - 13, 15 - 20, and 22 - 23** above, further in view of Deutsch (Deutsch, Alin; Fernandez, Mary; Suciu, Dan; "Storing Semistructured Data with STORED", 1999, Proceedings of the 1999 ACM SIGMOD international conference on management of data).

18.1. Manning as modified by Morgenstern teaches the persistence engine as recited in claims **1 - 3, 7 - 13, 15 - 20, and 22 - 23** above.

18.2. Claim 24 is a dependent claim of claim 23, and thereby inherits all of the rejected limitations of claim 23

18.3. Claim 28 is a dependent claim of claim 23, and thereby inherits all of the rejected limitations of claim 23.

18.4. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)*.

18.5. The art of Deutsch is directed toward a database for semistructured data, including XML *(Abstract)*.

18.6. Regarding claim 24, Manning as modified by Morgenstern does not specifically teach that the transform generator produces a transform that remodels the persistent data to approximate as closely as possible a preexisting transform from the set of transforms.

18.7. Regarding claim 28, Manning as modified by Morgenstern does not specifically teach that the transform generator produces a transform to remodel the persistent data to maximize data compression.

18.8. Regarding claim 24, Manning as modified by Morgenstern appears to teach that the transform generator produces a transform that remodels the persistent data to approximate as closely as possible a preexisting transform from the set of transforms *(first page, right-side column, fourth paragraph that starts with "In the first application . . ."; second page, left-side column, second paragraph, and third paragraph, and fourth paragraph, bullet points)*.

18.9. Regarding claim 28, Deutsch appears to teach that the transform generator produces a transform to remodel the persistent data to maximize data compression *(first page, right-side column, fourth paragraph that starts with "In the first application . . ."; second page, left-side column, first paragraph, the sentence that starts with, "The meaning of "good" depends on the application, but usually includes minimizing disk space . . . "; and second page, left-side column, second paragraph and third paragraph and fourth paragraph)*.

18.9.1. Regarding *(first page, right-side column, fourth paragraph that starts with "In the first application . . ."; second page, left-side column, first paragraph, the sentence that starts with, "The meaning of "good" depends on the application, but usually includes minimizing disk space . . . "; and second page, left-side column, second paragraph and third paragraph and fourth paragraph)*; it would have been obvious to design the transform generator to produce a transform to remodel the persistent data to maximize data compression.

18.10. The art of Manning as modified by Morgenstern and the art of Deutsch are analogous art because they both contain the problem storing XML data in a database.

18.11. The motivation to use the art of Deutsch with the art of Manning as modified by Morgenstern would have been obvious given the requirement recited in Deutsch of the need to generate a good relational schema *(first page, right-side column, last sentence, continuing on the second page)*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Deutsch with the art of Manning as modified by Morgenstern to produce the claimed invention.

19. **Claims 29 and 36** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning (U.S. Patent Application Publication Number US 2002/0103829) in view of Morgenstern (U.S. Patent No. 5,970,490).

19.1. **Regarding claim 29,** Manning appears to teach:

19.1.1. a communications interface (*paragraph [0038], "a file server providing access to the programs via a network transmission line, wireless transmission media, signals propagating through space, etc."*);

19.1.2. a data model description receiver to receive a data model description (*figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004]; and paragraph [0028]; please note that the DTD is included with the received document; it would have been obvious that persistent data from vector graphics had a different data model than e-commerce transactions)* from one of a plurality of different software components (*paragraph [0004] first and second sentences; it would have been obvious that the multiple data types recited, such as vector graphics and e-commerce transactions, were produced by different software components, wherein a software component is interpreted to include an application*), the software components having persistent data in accordance with different data models (*paragraph [0004] first and second sentences; it would have been obvious that persistent data from vector graphics had a different data model than e-commerce transactions*).

19.1.3. a set of transforms (*paragraph [0022]; it would have been obvious that a relational database has schemas, which are a set of transforms*).

19.1.4. a transform generator, operational during runtime of the apparatus, having an assembler to produce a transform based on the data model description independent of the software component from which the data model description was received (*figure 3, elements 102 - 110; and paragraph [0041]; and paragraphs [0028] and [0029]; since database schemas are produced during the runtime, it would have been obvious that there is a transform generator having an assembler*).

19.1.5. a storage device (*paragraph [0038]*);

19.1.6. a transform engine to apply a transform to format persistent data for storage from the format of the software component into a storage format that is compatible with a storage device independent of the software component (*figure 3, element 124, since the accessed object is stored, it would have been obvious that a transform is applied; and paragraph [0029], since each object (e.g. attribute value or content) is stored in an element table, it would have been obvious that a transform is applied to format the persistent data for storage; also paragraph [0022], paragraphs [0027] - [0029], and paragraph [0034]; it would have been obvious that the persistent data such as vector graphics and e-commerce transactions is being stored in a relational database, which is a storage format that is compatible with the receiving system and with a storage device independent of the software component*).

19.1.7. Manning does not specifically teach:

19.1.8. a data model comparator to produce a comparison independent of the software component from which the data model description is received between the data model description and a data model in a transform in the set of transforms.

19.1.9. a transform generator, operational during runtime of a system, having an assembler to produce a transform based on the data model description *and the comparison* independent of the software component from which the data model description was received.

19.1.10. Morgenstern appears to teach:

19.1.11. a data model comparator to produce a comparison independent of the software component from which the data model description is received between the data model description and a data model in a transform in a set of transforms (*figure 2, elements 22, 24, 32, 52, 36, 56, 42, 30; since the elements compare the schemas for different source and target data sources, it would have been obvious that element 30 is a data model comparator to produce a comparison independent of the software component from which the data model description is received between the data model description and a data model in a transform in a set of transforms*).

19.1.12. a transform generator, having an assembler to produce a transform based on the data model description *and the comparison* independent of the software component from which the data model description was received (*figure 2, elements 42, 22, 24, 32, 52, 36, 56, 30*).

19.2. Regarding claim 36, Manning appears to teach:

19.2.1. receiving a data model description (*figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004]; and paragraph [0028]; please note that the DTD is included with the received document)* from one of a plurality of different software components (*paragraph [0004] first and second sentences; it* *would have been obvious that the multiple data types recited, such as vector graphics and e-commerce transactions, were produced by different software components, wherein a software component is interpreted to include an application*), the software components having persistent data in accordance with different data models (*paragraph [0004] first and second sentences; it would have been obvious that persistent data from vector graphics is in a different data model than e-commerce transactions*), the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component (*paragraph [0004]*), the persistent data being associated with metadata describing the persistent data (*paragraph [0004]*) including a class file with bytecodes for each object (*paragraphs [0004] - [0005], it would have been obvious that a DTD was a class file with bytecodes for each document object*).

19.2.2. comparing the data model description to a preexisting data model independent of the software component from which the data model description is received (*figure 3, items 100 - 102; and paragraph [0028]*).

19.2.2.1. Regarding (*figure 3, items 100 - 102; and paragraph [0028]*); it would have been obvious that the data model description is compared to a preexisting data model since it is determined whether there are tables for the received DTD.

19.2.3. assembling a transform independent of the software component from which the data model description is received based on the data model description to establish a storage format for persistent data during runtime of the system (*paragraph [0028]; paragraph [0041]; figure 3, elements 102 - 110*).

19.2.4. applying a transform to format persistent data for storage from the format of the software component into a storage format that is compatible with a storage device independent of the software component (*figure 3, element 124, since the accessed object is stored, it would have been obvious that a transform is applied; and paragraph [0029], since each object (e.g. attribute value or content) is stored in an element table, it would have been obvious that a transform is applied; also paragraph [0022], paragraphs [0027] - [0029], and paragraph [0034]; it would have been obvious that the persistent data such as vector graphics and e-commerce transactions is being stored in a relational database, which is a storage format that is compatible with the receiving system and with a storage device independent of the software component*).

19.3. Manning does not specifically teach:

19.4. assembling a transform based on the data model description *and the comparison* to establish a storage format for persistent data.

19.5. Morgenstern appears to teach:

19.6. assembling a transform based on the comparison to establish a storage format for persistent data (*figure 2, elements 22, 24, 32, 52, 36, 56, 42, 30; since the elements compare the schemas for different source and target data sources, it would have been obvious that transform is assembled based on a comparison*).

19.7. The motivation to use the art of Morgenstern with the art of Manning would have been the several benefits recited in Morgenstern, including that self-description information simplifies the management of generated source code and the resulting compiled modules, which is especially useful in large systems (*column 6, lines 33 - 37, lines 1 - 2, lines 9 - 12*), and the advantage (*column 46, lines 41 - 45*) that the data transformation approach allows rules to be more declarative in nature, and also supports asynchronous processing of transformations, thereby being amenable to parallelization (*column 46, lines 35 - 41*), which would have been recognized as an advantage by the ordinary artisan.

19.8. Therefore, as discussed above, it would have been obvious to the ordinary artisan to use the art of Morgenstern with the art of Manning to produce the claimed invention.

20. **Claim 30** is rejected.under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 29 and 36** above, further in view of DeltaXML (web page for DeltaXML.com from September 2001 using www.archive.org at web.archive.org/web/20011021144026/www.deltaxml.com/prod-xmlschema-1000.html).

20.1. Manning as modified by Morgenstern teaches the data model receiver as recited in **claims 29 and 36** above.

20.2. Claim 30 is a dependent claim of claim 29, and thereby inherits all of the rejected limitations of claim 29.

20.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)*.

20.4. The art of DeltaXML is directed toward comparing XML schema DTD files to determine differences *(page 1, box labeled Description)*.

20.5. Regarding claim 30, Manning appears to teach a data model parser coupled to the assembler *(figure 3, elements 102 - 110; and paragraphs [0027], [0028])*.

20.6. Regarding claim 30, Manning as modified by Morgenstern does not specifically teach a data model *variance calculator* coupled to the assembler.

20.7. DeltaXML teaches a data model comparator *(page 1, box labeled Description)*, which also calculates the data model variance.

20.8. The art of DeltaXML and the art of Manning as modified by Morgenstern are analogous art because they both contain the problem of determining whether a pair of DTD's are different *(Manning, lines 14 - 17 of paragraph [0028])*.

20.9. The motivation to use the art of DeltaXML with the art of Manning as modified by Morgenstern would have been obvious given the need recited in Manning to determine whether documents have different DTD's *(Manning, lines 14 - 17 of paragraph [0028])*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of DeltaXML with the art of Manning as modified by Morgenstern to produce the claimed invention.

21. **Claims 31, 37 and 38** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 29 and 36** above, further in view of Nestorov (Nestorov,

Svetlozar; Abiteboul, Serge; Motwani, Rajeev; "Extracting Schema from Semistructured Data", 1998, Proceedings of the 1998 ACM SIGMOD international conference on Management of data).

21.1. Manning as modified by Morgenstern teaches the data model receiver as recited in claims 29 and 36 above.

21.2. Claim 31 is a dependent claim of claim 29, and thereby inherits all of the rejected limitations of claim 29.

21.3. Claim 37 is a dependent claim of claim 36, and thereby inherits all of the rejected limitations of claim 36.

21.4. Claim 38 is a dependent claim of claim 36, and thereby inherits all of the rejected limitations of claim 36.

21.5. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Title and Abstract; and paragraph [0020] regarding the XML document)*.

21.6. The art of Nestorov is directed toward extracting a schema (i.e. data model) from semistructured data (e.g. XML data) *(Title and Abstract)*.

21.7. Regarding claim 31, Manning appears to teach a data model parser coupled to the assembler (*figure 3, elements 102 - 110; and paragraphs [0027], [0028]*).

21.8. Regarding claim 31, Manning as modified by Morgenstern does not specifically teach a data model *approximator* coupled to the assembler.

21.9. Regarding claim 37, Manning as modified by Morgenstern does not specifically teach that assembling a transform includes measuring a variance between the data model description and a preexisting data model.

21.10. Regarding claim 38, Manning as modified by Morgenstern does not specifically teach that assembling a transform includes approximating a preexisting data model.

21.11. Regarding claim 31, Nestorov teaches a data model approximator *(page 1, Abstract; and page 6, section 3 Method Summary, first sentence)*.

21.12. Regarding claim 37, Nestorov teaches that assembling a transform includes measuring a variance between the data model description and a preexisting data model *(page 8 - 10, section 5.2 Distance function between types)*.

21.12.1. Regarding *(page 8 - 10, section 5.2 Distance function between types)*; it would have been obvious that assembling a transform includes measuring a variance between the data model description and a preexisting data model.

21.13. Regarding claim 38, Nestorov teaches that assembling a transform includes approximating a preexisting data model *(page 1, Abstract; and page 6, section 3 Method Summary, first sentence)*.

21.13.1. Regarding *(page 1, Abstract; and page 6, section 3 Method Summary, first sentence)*; it would have been obvious that assembling a transform includes approximating a preexisting data model.

21.14. The art of Nestorov and the art of Manning as modified by Morgenstern are analogous art because they both contain the problem of determining the data model of semistructured data.

21.15. The motivation to use the art of Nestorov with the art of Manning as modified by Morgenstern would have been obvious given the benefit recited in Nestorov of determining the data model for semistructured data where the data model is implicit is the data ***(Abstract)***. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Nestorov with the art of Manning as modified by Morgenstern to produce the claimed invention.

22. **Claims 34 and 41** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 29 and 36** above, further in view of Deutsch (Deutsch, Alin; Fernandez, Mary; Suciu, Dan; "Storing Semistructured Data with STORED", 1999, Proceedings of the 1999 ACM SIGMOD international conference on management of data).

22.1. Manning as modified by Morgenstern teaches the data model receiver as recited in **claims 29 and 36** above.

22.2. Regarding claim 34, Manning appears to teach a data model parser coupled to the assembler (*figure 3, elements 102 - 110; and paragraphs [0027], [0028]*).

22.3. Regarding claim 34, Manning as modified by Morgenstern does not specifically teach a data compression maximizer coupled to the assembler.

22.4. Regarding claim 41, Manning as modified by Morgenstern does not specifically teach that assembling a transform includes maximizing data compression.

22.5. Regarding claim 34, Deutsch appears to teach a data compression maximizer *(second page, left-side column, first paragraph, the sentence that starts with, "The meaning of "good" depends on the application, but usually includes minimizing disk space . . . ")*.

22.5.1. Regarding *(second page, left-side column, first paragraph, the sentence that starts with, "The meaning of "good" depends on the application, but usually includes minimizing disk space . . . ")*; it would have been obvious to use a data compression maximizer.

22.6. Regarding claim 41, Deutsch appears to teach that assembling a transform includes maximizing data compression *(second page, left-side column, first paragraph, the sentence that starts with, "The meaning of "good" depends on the application, but usually includes minimizing disk space . . . ")*.

22.6.1. Regarding *(second page, left-side column, first paragraph, the sentence that starts with, "The meaning of "good" depends on the application, but usually includes minimizing disk space . . . ")*; it would have been obvious that assembling a transform includes maximizing data compression.

22.7. The art of Manning as modified by Morgenstern and the art of Deutsch are analogous art because they both contain the problem storing XML data in a database.

22.8. The motivation to use the art of Deutsch with the art of Manning as modified by Morgenstern would have been obvious given the requirement recited in Deutsch of the need to generate a good relational schema *(first page, right-side column, last sentence, continuing on the second page)*.

22.9. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Deutsch with the art of Manning as modified by Morgenstern to produce the claimed invention.

23. Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 29 and 36** above, further in view of Mani (U.S. Patent 6,654,734 B1).

23.1. Manning as modified by Morgenstern teaches the data model description receiver as recited in **claims 29 and 36** above.

23.2. Claim 35 is a dependent claim of claim 29, and thereby inherits all of the rejected limitations of claim 29.

23.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Title and Abstract; and paragraph [0020] regarding the XML document)*.

23.4. The art of Mani is directed toward a method for query optimization for XML document databases *(Title and Abstract)*.

23.5. Manning appears to teach a data model parser coupled to the assembler (*figure 3, elements 102 - 110; and paragraphs [0027], [0028]*).

23.6. Manning as modified by Morgenstern does not specifically teach *an indexing estimator* coupled to the assembler.

23.7. Mani appears to teach an indexing estimator *(column 11, lines 54 -57, the referenced index access cost estimator)*.

23.7.1. Regarding *(column 11, lines 54 -57, the referenced index access cost estimator)*; it would have been obvious to use an indexing estimator.

23.8. The art of Mani and the art of Manning as modified by Morgenstern are analogous art because they both contain the problem of queries for an XML database *(Mani, Title)* and *(Manning, paragraph [0030])*.

23.9. The motivation to use the art of Mani with the art of Manning as modified by Morgenstern would have been obvious given the benefit recited in Mani of query optimization *(Title and Abstract)*, which would have been recognized by the ordinary artisan as saving time in a query.

23.10. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Mani with the art of Manning as modified by Morgenstern to produce the claimed invention.

24. **Claims 32, 33, 39 and 40** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 29 and 36** above, further in view of Ives (Ives, Zachary G.; Florescu, Daniela; Friedman, Marc; Levy, Alon; Weld, Daniel S.; "An Adaptive Query Execution System for Data Integration", 1999, SIGMOD 1999).

24.1. Manning as modified by Morgenstern teaches a data model description receiver as recited in **claims 29 and 36** above.

24.2. Claim 32 is a dependent claim of claim 29, and thereby inherits all of the rejected limitations of claim 29.

24.3. Claim 33 is a dependent claim of claim 32, and thereby inherits all of the rejected limitations of claim 32.

24.4. Claim 39 is a dependent claim of claim 36, and thereby inherits all of the rejected limitations of claim 36.

24.5. Claim 40 is a dependent claim of claim 36, and thereby inherits all of the rejected limitations of claim 36.

24.6. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Title and Abstract; and paragraph [0020] regarding the XML document)*.

24.7. The art of Ives is directed toward an adaptive query execution system for data integration *(Title)*.

24.8. Regarding claims 32, Manning appears to teach a data model parser coupled to the assembler (*figure 3, elements 102 - 110; and paragraphs [0027], [0028]*).

24.9. Regarding claim 32, Manning as modified by Morgenstern does not specifically teach an *efficient storage/retrieval speed maximizer* coupled to the assembler.

24.10. Regarding claim 33, Manning as modified by Morgenstern does not specifically teach an efficient storage/retrieval speed maximizer comprising a *read/write iterator*.

24.11. Regarding claim 39, Manning as modified by Morgenstern does not specifically teach that assembling a transform includes maximizing data storage speed and/or data retrieval speed.

24.12. Regarding claim 40, Manning as modified by Morgenstern does not specifically teach that maximizing speed includes iteratively performing data read/write trials and selecting the fastest trial.

24.13. Regarding claim 32, Ives appears to teach an efficient storage/retrieval speed maximizer *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*.

24.13.1. Regarding *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*; it would have been obvious to use an efficient storage/retrieval speed maximizer.

24.14. Regarding claim 33, Ives appears to teach an efficient storage/retrieval speed maximizer comprising a read/write iterator *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*.

24.14.1. Regarding *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*; it would have been obvious to use an efficient storage/retrieval speed maximizer comprising a read/write iterator.

24.15. Regarding claim 39, Ives appears to teach that assembling a transform includes maximizing data storage speed and/or data retrieval speed *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*.

24.15.1. Regarding *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*; it would have been obvious that assembling a transform includes maximizing data storage speed and/or data retrieval speed.

24.16. Regarding claim 40, Manning appears to teach that maximizing speed includes iteratively performing data read/write trials and selecting the fastest trial *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*.

24.16.1. Regarding *(page 304, first paragraph, the sentence that starts with "The query execution . . . ")*; it would have been obvious that maximizing speed includes iteratively performing data read/write trials and selecting the fastest trial.

24.17. The art of Manning as modified by Morgenstern and the art of Ives are analogous art because they are both contain the problem of data queries *(Manning, paragraph [0030])* and Ives *(Title)*.

24.18. The motivation to use the art of Ives with the art of Manning as modified by Morgenstern would have been obvious given the statement recited in Ives that it is important to optimize the time to initial answers to a query *(page 300, left-side column, the paragraph that starts with "Since data integration . . .")*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Ives with the art of Manning as modified by Morgenstern to produce the claimed invention.

25. **Claim 42** is rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Morgenstern as applied to **claims 29 and 36** above, further in view of Schoning (Schoning, Harald; "Tamino - a DBMS Designed for XML", 2001 Proceedings 17th International Conference on Data Engineering, 2-6 April 2001).

25.1. Manning as modified by Morgenstern teaches receiving a data model description as recited in **claims 29, 34, 36 and 41** above.

25.2. Claim 42 is a dependent claim of claim 36, and thereby inherits all of the rejected limitations of claim 36.

25.3. The art of Manning as modified by Morgenstern is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)*.

25.4. The art of Schoning is directed toward a database management system designed for XML *(Title)*.

25.5. Manning as modified by Morgenstern does not specifically teach that the transform generator produces a transform to remodel the persistent data to maximize efficient retrieval for an application.

25.6. Regarding claim 42, Schoning appears to teach that assembling a transform includes optimizing efficient indexing for the persistent data *(page 152, section labeled "Indexing and storage methods")*.

25.6.1. Regarding *(page 152, section labeled "Indexing and storage methods")*; it would have been obvious that assembling a transform includes optimizing efficient indexing for the persistent data.

25.7. The art of Manning as modified by Morgenstern and the art of Schoning are analogous art because they are both directed to the art of XML databases.

25.8. The motivation to use the art of Schoning with the art of Manning as modified by Morgenstern would have been obvious given the statement recited in Schoning that indexes are indispensable in database systems because otherwise large amounts of data could not be efficiently queried *(page 152, section labeled "Indexing and storage methods")*. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of Schoning with the art of Manning as modified by Morgenstern to produce the claimed invention.

26. **Claims 43 - 44 and 47 - 49** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning (U.S. Patent Application Publication Number US 2002/0103829) in view of Official Notice.

26.1. Regarding claim 43, Manning appears to teach:

26.1.1. a machine-readable medium comprising instructions that are executed by a machine *(paragraphs [0021] and [0022])*.

26.1.1.1. Regarding *(paragraphs [0021] and [0022])*; it would have been obvious that instructions cause a machine to execute because a computer was a machine, and computers execute instructions.

26.1.2. receiving persistent data having a model structure *(figure 3, item 100 - 102; please note that a DTD is a Document Type Definition that provides attributes for each element in the document, and indicates the relationship of the elements - please refer to paragraph [0004]; and paragraph [0028]; please note that the DTD is included with the received document)* from one of a plurality of different software components (*paragraph [0004] first and second sentences; it would have been obvious that the multiple data types recited, such as vector graphics and e-commerce transactions, were produced by different software components, wherein a software component is interpreted to include an application*), the software components having persistent data in different model structures (*paragraph [0004] first and* *second sentences; it would have been obvious that persistent data from vector graphics is in a different model structure than e-commerce transactions*), the persistent data relating to diverse types of objects constructed at runtime of the software component and needed during more than one invocation of the software component (*paragraph [0004]*).

26.1.3. receiving metadata comprising at least in part a description of the model structure (*figure 3, items 100 - 102; and paragraph [0028]*), the metadata describing the persistent data (*paragraph [0004]*) including a class file with bytecodes for each object (*paragraphs [0004] - [0005], it would have been obvious that a DTD was a class file with bytecodes for each document object*).

26.1.4. establish, using the metadata and without the using the software component from which the persistence package was received, during a runtime of the machine, a storage format for the persistent data (*paragraph [0028]; paragraph [0041]; figure 3, elements 102 - 110; paraphrasing a definition of "format" from the IBM Dictionary of Computing, a "format" is a specified arrangement of fields, and it would have been obvious to the ordinary artisan that a storage format is established*).

26.1.5. apply the established storage format to the persistent data to format the persistent data for storage (*figure 3, element 124, since the accessed object is stored, it would have been obvious that the established storage format is applied; and paragraph [0029], since each object (e.g. attribute value or content) is stored in an element table, it would have been obvious that an established storage format is applied*) from the format of the software component into a storage format that is compatible with the machine and with a storage device independent of the software component (*paragraph [0022], paragraphs [0027] - [0029], and paragraph [0034]; it would have been obvious that the received data such as vector graphics and e-commerce transactions is being stored in a relational database,* *which is a storage format that is compatible with the receiving system and with a storage device independent of the software component*).

26.1.6. Manning does not specifically teach:

26.1.6.1. the software components having persistent *data in different model structures*.

26.1.7. Official Notice is taken that it was old and well known in the art that vector graphics was in a different format than data from e-commerce transactions, and therefore, the software components had persistent *data in different model structures*. Also, please refer to the art cited in the Conclusion section of this Office Action regarding the format of e-commerce data and vector graphics data.

26.1.8. The motivation to use Official Notice with the art of Manning would have been the knowledge of the ordinary artisan that a relational database such as that used in Manning would allow fast searching of large amounts of data, which the ordinary artisan would have recognized as a benefit that saves time and allows analysis of data.

26.1.9. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use Official Notice with the art of Manning to produce the claimed invention.

26.2. Regarding claim 44, Manning appears to teach instructions *(paragraphs [0021] and [0022])*, that when executed, cause a machine to store the persistent data using the storage format *(figure 3, item 124; and paragraph [0029])*.

26.3. Regarding claim 47, Manning appears to teach instructions, that when executed cause a machine to retrieve the persistent data using the storage format *(figure 4, all items; and paragraph [0030])*.

26.4. Regarding claim 48, Manning appears to teach instructions, that when executed, cause a machine to select and/or create, based on the metadata, a transform to establish at least one of the storage format and the storage location *(figure 3, items 102 - 110; and paragraph [0028], sentences 1 - 4)*.

26.5. Regarding claim 49, Manning appears to teach receiving persistent data compatible with one of any type of processor, any type of programming language, any type of operating system, and any type of architecture *(paragraph [0021])*.

27. **Claims 45 and 46** are rejected under 35 U.S.C. 103(a) as being unpatentable over Manning in view of Official Notice as applied to claims **43 - 44 and 47 - 49** above, in view of XML ("Extensible Markup Language (XML) 1.0"; W3C Recommendation 10-Feb-98, 1998).

27.1. Manning as modified by Official Notice teaches receiving persistent data having a model structure as recited in claims **43 - 44 and 47 - 49** above.

27.2. Claim 45 is a dependent claim of claim 43, and thereby inherits all of the rejected limitations of claim 43.

27.3. Claim 46 is a dependent claim of claim 45, and thereby inherits all of the rejected limitations of claim 45.

27.4. The art of Manning as modified by Official Notice is directed toward a method, system, program, and data structures for managing structured XML documents in a database *(Manning, Title and Abstract; and paragraph [0020] regarding the XML document)*.

27.5. The art of XML is directed toward describing the Extensible Markup Language (XML) *(Abstract)*.

27.6. Regarding claim 45, Manning appears to teach receiving metadata *(figure 3, item 100; and paragraph [0004] - please note that an XML document contains both persistent data and metadata)*.

27.7. Regarding claim 46, Manning appears to teach receiving a persistence package comprising persistent data and metadata *(figure 3, item 100; and paragraph [0004] - please note that an XML document contains both persistent data and metadata)*, and to extract the persistent data and the metadata from the persistence package (*paragraphs [0028] and [0029]; and figure 3, all elements*).

27.8. Regarding claim 45, Manning as modified by Official Notice does not specifically teach receiving metadata *conforming to a metadata template comprising rules for describing a data model structure of the persistent data*.

27.9. Regarding claim 45, Manning appears to teach that the metadata received in claim 45 conforms to a metadata template comprising rules for describing a data model structure of the persistent data *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*.

27.9.1. Regarding *(page 2, section 2. Documents, first sentence; and page 3, section 2.1 Well-Formed XML Documents)*; the reference XML describes the rules that the metadata conforms to, and specifically the production in section 2.1 is a metadata template.

27.10. The art of XML and the art of Manning are analogous art because they both contain the art of interpreting XML documents.

27.11. The motivation to use the art of XML with the art of Manning would have been obvious given the need in Manning to interpret XML documents, and the rules given in XML to form valid XML documents.

27.12. Therefore, as discussed above, it would have been obvious to the ordinary artisan at the time of invention to use the art of XML with the art of Manning to produce the claimed inventions.

28. **Examiner's Note:** Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

Conclusion

29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Russ Guill whose telephone number is 571-272-7955. The examiner can normally be reached on Monday - Friday 9:00 AM - 5:30 PM.

30. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached on 571-272-3753. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any inquiry of a general nature or relating to the status of this application should be directed to the TC2100 Group Receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Russ Guill
Examiner
Art Unit 2123

RG

PAUL RODRIGUEZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

1/17/07